Đã bắt đầu vào lúc	Thứ hai, 21 Tháng mười một 2022, 1:33 PM
Tình trạng	Đã hoàn thành
Hoàn thành vào lúc	Thứ hai, 21 Tháng mười một 2022, 2:50 PM
Thời gian thực hiện	1 giờ 16 phút
Điểm	3,00/3,00
Điểm	**10,00** của 10,00 (**100**%)

Câu hỏi 1

Chính xác

Điểm 1,00 của 1,00

Implement three following hashing function:

```
long int midSquare(long int seed);
long int moduloDivision(long int seed, long int mod);
long int digitExtraction(long int seed, int* extractDigits, int size);
```

Note that:

In midSquare function: we eliminate 2 last digits and get the 4 next digits.

In digitExtraction: extractDigits is a sorted array from smallest to largest index of digit in seed (index starts from 0). The array has size **size.**

For example:

Test	Result

`int a[]={1,2,5};`
`cout << digitExtraction(122443,a,3);`
	223

`cout <<midSquare(9452);`	3403

Answer: (penalty regime: 0, 0, 0 %)

Reset answer

```
long int midSquare(long int seed)
{
    long int m = seed*seed;
    m = m/100;
    return (m%10000);

}
long int moduloDivision(long int seed, long int mod)
{
    return (seed%mod);
}
long int digitExtraction(long int seed,int* extractDigits,int size)
{
    string s = to_string(seed);
    long int ans = 0;
    for(int i = 0; i < size; i++){
        int index = extractDigits[i];
        ans = ans*10+(s[index]-'0');
    }
    return ans;
}
```

	Test	Expected	Got	

✓	`int a[]={1,2,5};`
	`cout << digitExtraction(122443,a,3);`
		223	223	✓

✓	`cout <<midSquare(9452);`	3403	3403	✓

Passed all tests! ✓

Chính xác

Điểm cho bài nộp này: 1,00/1,00.

Câu hỏi 2

Chính xác

Điểm 1,00 của 1,00

Implement function

```
int foldShift(long long key, int addressSize);
int rotation(long long key, int addressSize);
```

to hashing key using Fold shift or Rotation algorithm.

Review Fold shift:

The **folding method** for constructing hash functions begins by dividing the item into equal-size pieces (the last piece may not be of equal size). These pieces are then added together to give the resulting hash value.

For example:

Test	Result
`cout << rotation(600101, 2);`	26

Answer: (penalty regime: 0 %)

Reset answer

```
int foldShift(long long key, int addressSize)
{
string s = to_string(key);
    //s = to_string(lastDigit) + s;
    int len = s.length();
    int loop = len/addressSize;
    int r = len%addressSize;
    int i = 0;
    int ans = 0;
    for(int j = 0; j < loop; j++){
        string tmp = s.substr(i,addressSize);
        ans += stoi(tmp);
        i += addressSize;
    }
    //i-= addressSize;
    if(r >0){
        string tmp = s.substr(i,r);
        ans += stoi(tmp);
    }
    string resault = to_string(ans);
    resault = resault.substr(int(resault.length())-addressSize,addressSize);
    return stoi(resault);
}

int rotation(long long key, int addressSize)
{
        int lastDigit = key%10;
    key = key/10;
    string s = to_string(key);
    s = to_string(lastDigit) + s;
    int len = s.length();
    int loop = len/addressSize;
    int r = len%addressSize;
    int i = 0;
    int ans = 0;
    for(int j = 0; j < loop; j++){
        string tmp = s.substr(i,addressSize);
        ans += stoi(tmp);
        i += addressSize;
    }
    //i-= addressSize;
    if(r >0){
        string tmp = s.substr(i,r);
        ans += stoi(tmp);
    }
    string resault = to_string(ans);
    resault = resault.substr(int(resault.length())-addressSize,addressSize);
    return stoi(resault);

```

	Test	Expected	Got	
✓	`cout << rotation(600101, 2);`	26	26	✓

Passed all tests! ✓

Chính xác

Điểm cho bài nộp này: 1,00/1,00.

Câu hỏi 3

Chính xác

Điểm 1,00 của 1,00

There are `n` people, each person has a number between `1` and `100000` (`1 ≤ n ≤ 100000`). Given a number `target`. Two people can be matched as a **perfect pair** if the sum of numbers they have is equal to `target`. A person can be matched no more than 1 time.

Request: Implement function:

```
int pairMatching(vector<int>& nums, int target);
```

Where `nums` is the list of numbers of `n` people, `target` is the given number. This function returns the number of **perfect pairs** can be found from the list.

Example:

The list of numbers is `{1, 3, 5, 3, 7}` and `target = 6`. Therefore, the number of **perfect pairs** can be found from the list is `2` (pair `(1, 5)` and pair `(3, 3)`).

Note:

In this exercise, the libraries `iostream, string, cstring, climits, utility, vector, list, stack, queue, map, unordered_map, set, unordered_set, functional, algorithm` has been included and `namespace std` are used. You can write helper functions and classes. Importing other libraries is allowed, but not encouraged, and may result in unexpected errors.

For example:

Test	Result

`vector<int>items{1, 3, 5, 3, 7};`
`int target = 6;`
`cout << pairMatching(items, target);`
	2

`int target = 6;`
`vector<int>items{4,4,2,1,2};`
`cout << pairMatching(items, target);`
	2

Answer: (penalty regime: 0, 0, 0, 5, 10, ... %)

Reset answer

```
#include <bits/stdc++.h>
int pairMatching(vector<int>& nums, int target) {
    vector<int> tmp = nums;
    sort(tmp.begin(),tmp.end());
    int size = int(tmp.size());
    int lo = 0;
    int hi = size-1;
    int count = 0;
    while(lo < hi){
        int sum = tmp[lo]+tmp[hi];
        if(sum == target){
            lo++;
            hi--;
            count++;
        }
        if(sum > target){
            hi--;
        }
        if(sum < target){
            lo++;
        }
    }
    return count;
}
```

	Test	Expected	Got	

✓	`vector<int>items{1, 3, 5, 3, 7};`
	`int target = 6;`
	`cout << pairMatching(items, target);`
		2	2	✓

Passed all tests! ✓

Chính xác

Điểm cho bài nộp này: 1,00/1,00.

BÁCH KHOA E-LEARNING

WEBSITE

HCMUT

MyBK

BKSI

LIÊN HỆ

268 Lý Thường Kiệt, P.14, Q.10, TP.HCM

☎ (028) 38 651 670 - (028) 38 647 256 (Ext: 5258, 5234)

✉ elearning@hcmut.edu.vn